Driving Scene Understanding: How much temporal context and spatial resolution is necessary?

Ramashish Gaurav[†,*], Bryan Tripp[†], Apurva Narayan[†, ‡]
[†] University of Waterloo
[‡] University of British Columbia

Abstract

Driving Scene Understanding is a broad field which addresses the problem of recognizing a variety of on-road situations; namely driver behaviour/intention recognition, driver-action causal reasoning, pedestrians' and nearby vehicles' intention recognition, etc. Many existing works propose excellent AI based solutions to these interesting problems by leveraging visual data along with other modalities. However, very few researchers venture into determining the necessary metadata of the visual inputs to their models. This work attempts to put forward some useful insights about the required spatial resolution and temporal context/depth of the visual data for Driving Scene Understanding.

Keywords: driving scene understanding, 3D-CNNs, spatial resolution, temporal depth

1. Introduction

With the ubiquitous use of AI in every walk of life, the transportation domain is not untouched by it; AI is increasingly finding its varied usage in autonomous and semi-autonomous vehicles. In a future of hybrid transportation where more and more of human-driven vehicles and self-driven vehicles would share the road, there arises a need to recognize a driving scene for better contextual communication among vehicles to make informed driving decisions, thereby, increasing the on-road safety. For the same, researchers employ a variety of modalities e.g. visual data, Controller Area Network (CAN) bus data, LiDAR data, GPS data, etc. to understand a driving scene, with the visual data being the primary modality.

1.1. Driving Scene Understanding

Towards the cause of Driving Scene Understanding, a number of works have been done; each of which addresses different subsets of problems in this broad domain. Authors in [1–3] leverage Hidden Markov Models for driver intention recognition. Recently, Casas et al. [4] put forward a fully convolutional neural network method for predicting the driving intent of other vehicles in the context of self-driving ones. For the task at hand, they leveraged the 3D point clouds produced by the mounted LiDAR and dynamic maps of environment containing lanes, intersections, etc. Frossard et al. [5] proposed the usage of a Convolutional-Recurrent architecture for detecting the turn signals and flashers in video sequences. Few [6, 7] have also attempted to understand and predict the pedestrian intentions to improve the Driver Intention Recognition systems. Torstensson et al. [8] proposed a Convolutional and LSTM based network to predict the actions of the in-vehicle driver. In a work related to Driving Action Anticipation, Aliakbarian et al. [9] introduced a new dataset: VIENA^2 and proposed a multi-modal LSTM based network to forecast driver actions from visual and sensor data.

A recently published dataset by Ramanishka et al. [10]: **Honda Research Institute Driving Dataset** (**HDD**) which they benchmark for a variety of driving scenes, has gained traction of late for the task of Driving Scene Understanding. Xu et al. [11] used this dataset to evaluate their new recurrent architecture for a variety of online action-detection tasks (including driving scenes). The authors in both the papers [10, 11] used a Convolutional network coupled with a Recurrent architecture for recognizing the driver actions from visual

[*] rgaurav@uwaterloo.ca

as well as the CAN bus sensor data. The HDD dataset was further studied by Li et al. [12] to identify the causal reasons for the human drivers to stop on-road. This dataset was also employed for interaction modeling between ego-car and other on-road objects (e.g. pedestrians, lanes, traffic light) by using Graph Convolutional Networks [13]. Owing to the popularity and variety, we use the HDD dataset along with VIENA^2 for our experiments.

1.2. **3D-CNNs**

As can be inferred from the above paragraphs, visual data forms an important modality and Convolutional networks are critical to learning spatial representations. While most of the works use pre-trained 2D-CNNs (primarily on ImageNet) to extract spatial features, followed by the usage of Recurrent networks for learning temporal dynamics; researchers haven't emphasized enough on 3D-CNNs based models for Driving Scene Understanding. 3D-CNNs based models can jointly learn the spatial and temporal representations in a video [14–16] and can also be used for human action recognition [17]. Another missing aspect of most of the works is the absence of insights in the necessary spatial and temporal resolution of visual inputs for Driving Scene Understanding. One can expect higher spatial resolution to be favourable performance-wise; however, it comes with an added cost of increased computational complexity. Increased prior temporal context to understand an ongoing driving scene might introduce irrelevant past contextual details (e.g. lane changes are quicker than U-turns). In addition, the joint contribution of higher spatial resolution and increased temporal context also poses hardware implications during training and deployment.

Therefore, in accordance with our *titled* problem statement, we build a C3D [14] inspired 3D-CNNs based network to investigate the degree of the required spatial resolution and temporal depth for Driving Scene Understanding. C3D is one of the best established architectures for a variety of video based tasks and our improved results with it shows its efficacy. Since Tran et al. [14] already did exhaustive hyper-parameter search while building their C3D architecture, we reuse their findings in building our model and focus on the less studied spatial and temporal resolution need instead.

1.3. **Contributions**

Our work's contribution is three fold, summarised below.

- We demonstrate superior results on the visual data alone with our C3D inspired 3D-CNNs based architecture
- Given a model, we attempt to identify the optimum spatial resolution and temporal context/depth of the input necessary for recognizing a variety of driving scenes (collectively) in general setting
- In a first, we introduce a new accuracy metric **ASiST@x** which jointly measures the accuracy of recognizing scenes within a certain time as well as the recognition continuity of ongoing scenes

We organize our paper as follows. In Section 2 we define the specifics of our experiments, followed by the experimental details in Section 3. We then present and analyse our results in Section 4 followed by consolidating our findings in the conclusion Section 5.

2. **Experiment Specifics**

In this section we formally describe the elements of our designed experiments. We begin by defining the term **Temporal Context/Depth**, followed by defining the term **Spatial Resolution**, and end this section with a description of our 3D-CNNs based model.

2.1. **Temporal Context/Depth**

The RGB video data is composed of a continuous sequence of frames f_m where $m \in [1, \cdots, N]$; N is the total number of frames in the video, usually at 30 FPS. We can label

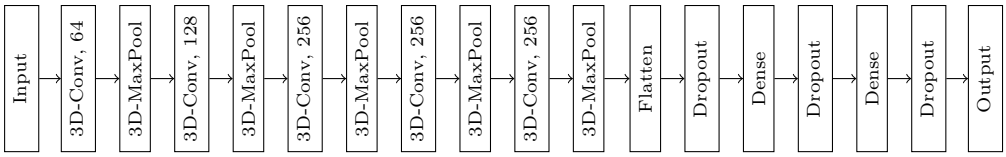


Figure 1. Our Model's Architecture; "3D-Conv, 64" implies 64 filters in the 3D-Convolutional layer

each of the frames f_m sequentially to denote various temporally arranged ongoing driving scenes. Let $F_{i,j}$ be a contiguous sequence of consecutive frames f_m where $m \in [i, \cdots, j]$. We therefore build a sliding window $F_{i,i+l-1}$ of l frames where the task is to predict the label of the last frame f_{i+l-1}; the label denotes the ongoing driving scene in the current frame f_{i+l-1} in the context of past $l-1$ frames. We slide the window one frame at a time. Contrary to [10, 11] where authors first construct a sequence of 90 frames and then predict the labels of each frame in one go, our formulation is more favourable and responsive to time critical on-road situations, as it recognizes the ongoing driving scene immediately upon arrival of a new frame. We mention this parameter l as the **Temporal Depth**. In our experiments, we consider three different values of $l \in \{16, 24, 32\}$; *what should be the optimum l?*

2.2. Spatial Resolution

Irrespective of the various Neural Network models to learn the spatial features, the question remains: *what should be the appropriate spatial resolution of the input frames?* Towards this cause, we resize the frames to varying spatial resolutions; low resolution: 36×64 pixels, medium resolution: 72×128 pixels, and high resolution: 108×192 pixels, and conduct extensive experiments. The first dimension corresponds to the number of rows and second dimension corresponds to the number of columns in the RGB frames.

2.3. 3D-CNN based Model

To benchmark our findings we build our model (Figure 1) based on the C3D architecture proposed by Tran et al. [14]. Each of the Convolutional layers has a kernel size of $(3, 3, 3)$. The Max-Pooling size is set to $(2, 2, 2)$ except for the first pooling layer where it is set to $(1, 2, 2)$. The first dimension of the kernel size and pool size correspond to the temporal dimension, last two dimensions correspond to the spatial dimension. The strides for the Convolutional layers are set to $(1, 1, 1)$ and that for the Max-Pooling layers are set as the pool size. The number of neurons in each of the non-output Dense layers is set to 2048. All the neurons in our model are $ReLU$ neurons, except for the output layer which has softmax activation. To prevent overfitting, we $L2$ regularize the kernels and keep the dropout probability $= 0.25$. The learning rate is fixed at 0.0001 and we use the optimizer Adam [18] to train our network. In accordance with [10, 11] we also use Focal Loss [19] (with $\gamma = 2.0$) as the loss function to account for the class imbalance problem in the HDD dataset.

3. Experiments

In this section we describe the details of our conducted experiments. We begin by a short introduction of the VIENA^2 and the HDD dataset, followed by the experiment methodology. Note that we used only the visual data, and sampled the frames at 3 FPS for both datasets (authors in [10, 11, 13] sample the HDD dataset at 3 FPS).

3.1. VIENA^2 Dataset

The VIENA^2 dataset consists of multiple 5 seconds long labelled video clips (30 FPS, 1280×1920 pixels resolution, ≈ 8.5 hours total) along with the aligned sensor data (speed and steering angle) collected from the GTA V video game for five different driving Scenarios; namely (1) Driver Maneuvers (DM $\approx 2h45m$), (2) Accidents (AC $\approx 1h25m$), (3) Traffic Rules (TR $\approx 1h30m$), (4) Pedestrian Intentions (PI $\approx 1h10m$), and (5) Front Car Intentions (FCI $\approx 1h45m$). For each of the 5 scenarios, there are 3 different splits: *Daytime* split,

Table 1. Average run-time (rounded) in minutes of $E_{108\times192,16}$ for the HDD and VIENA^2 dataset. Note that the training was done in parallel on 4 GPUs, but the inference was done on a single GPU.

	HDD	**HDD**	**VIENA^2** - *Random* split				
	Layer 0	**Layer 1**	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**
Training	113	105	3	2	2	2	2
Inference	79	82	2	1	1	1	1

Table 2. VIENA^2 accuracy scores for all 5 Scenarios - *Random* split; **DM** - Driver Maneuvers, **AC** - Accidents, **TR** - Traffic Rules, **PI** - Pedestrian Intentions, **FCI** - Front Car Intentions. For respective class acronyms (e.g. FF, SS, AP, NP, etc.) definition, refer Section 2.1 of [9].

	Scenario 1: DM						**Scenario 2: AC**				**Scenario 3: TR**					**Scenario 4: PI**				**Scenario 5: FCI**					
	FF	**SS**	**LL**	**RR**	**CL**	**CR**	**AC**	**AP**	**AA**	**NA**	**SR**	**PR**	**WD**	**CD**	**DO**	**CR**	**SS**	**AS**	**NP**	**FF**	**SS**	**LL**	**RR**	**CL**	**CR**
$E_{36\times64,16}$	72.9	95.4	88.1	88.2	59.7	76.3	85.9	68.7	41.3	96.2	98.7	46.4	58.0	71.6	45.1	53.3	39.6	62.1	60.0	80.1	84.4	64.7	66.4	31.0	1.8
$E_{72\times128,16}$	83.2	95.7	84.7	83.9	**68.9**	**82.9**	81.0	55.4	50.8	95.9	**100**	50.4	57.7	69.0	36.9	52.1	36.8	**75.1**	67.0	80.4	88.6	70.8	66.4	36.8	0.0
$E_{108\times192,16}$	79.4	95.2	76.5	72.8	65.1	76.6	73.6	70.3	45.1	92.6	99.0	**54.6**	49.3	68.6	41.5	68.3	26.8	71.2	58.5	71.3	**92.8**	69.5	69.2	47.7	1.8
[9]	**88.0**	**97.2**	**95.8**	**90.4**	64.9	65.4	**86.1**	**80.5**	**80.2**	**100**	95.1	40.0	**75.0**	**85.7**	**48.6**	**78.2**	**76.6**	63.6	**74.1**	**91.2**	83.5	**84.6**	**81.4**	**59.4**	**66.8**

Weather split, and *Random* split; we use the *Random* split (70% training clips, 30% test clips). Note that due to only 5 seconds long clips, we could not experiment for $l = \{24, 32\}$.

3.2. Honda Research Institute Driving Dataset (HDD)

The HDD dataset [10] is a real world largest public dataset (till date [13]) containing 104 hours of egocentric driving data with per frame annotation. It has video data (30 FPS, 720×1280 pixels resolution) and CAN Bus sensor data for 137 driving sessions (avg. span 45 minutes); 100 sessions are used for training and rest for testing. This dataset has 4 "Layers" of annotation for Driving Scene Understanding; namely: Goal-Oriented Action, Stimulus-Driven Action, Cause, and Attention (authors in [10] use the term *"Layer"* to denote groups of semantically related driving scenes). We experiment on two annotation layers: **Layer 0 - Goal-Oriented Action** and **Layer 1 - Cause**. We do not use the CAN Bus sensor data.

3.3. Model Execution

We conduct a number of experiments with VIENA^2 (5 runs each) and both Layers of the HDD dataset (3 runs each), where each experiment corresponds to a combination of a spatial resolution and a temporal depth. Across all the experiments, the model hyper-parameters are kept constant for fair comparison; only the spatial resolution and the temporal depth is varied. For convenience, we mention each experiment as $E_{r\times c,l}$ where $r \times c \in \{36 \times 64, 72 \times 128, 108 \times 192\}$ pixels resolution, and $l \in \{16, 24, 32\}$ frames temporal depth. Thus, the shape of input to our model is $(batch\_size, l, r, c, 3)$ where 3 is the RGB channel dimension. The experiments are executed on nodes with 4 NVIDIA V100 32GB GPUs. For the HDD and VIENA^2 dataset, 1 epoch's training and inference run-time figures (averaged across runs) of the best performing experiment $E_{108\times192,16}$ are mentioned in Table 1. Since the authors [9–11, 13] (with whom we compare our results later) do not provide run-time estimates of their experiments, we are unable to present a comparison. Our code is publicly available [1].

4. Results & Analysis

Here, we present our results (averaged across runs) and analyse them to get insights in the optimal spatial and temporal resolution required for understanding a variety of driving scenes. We begin by proving the efficacy of 3D-CNNs over existing approaches, followed by analysing the per-frame **ASiST@x** plots of all $E_{r\times c,l}$ collectively on the HDD dataset.

4.1. VIENA^2 Dataset Results

We present our results for VIENA^2 dataset in Table 2, where we compare our class-wise accuracy scores (obtained with $E_{r\times c,16}$) with that of Aliakbarian et al. [9] (obtained on

[1] https://github.com/R-Gaurav/DSU-3D-CNNs

Table 3. Average Precision (AP) results for HDD Layer 0. **Itr. Pass.**: Intersection Passing; **Crs. Pass.**: Crosswalk Passing; **Rail. Pass.**: Railroad Passing; **mAP**: mean Average Precision

	AP results: Layer 0 - Goal Oriented Action Layer											**mAP**
Methods	**Right Turn**	**Itr. Pass.**	**Merge**	**Left Lane Change**	**Right Lane Branch**	**Right Lane Change**	**Left Turn**	**Crs. Pass.**	**Rail. Pass.**	**Left Lane Branch**	**U-Turn**	
[10]	54.43	65.74	4.86	27.84	1.77	26.11	57.79	**16.08**	2.56	25.76	13.65	26.96
[11]	57.3	63.5	3.5	28.4	**10.5**	37.8	57.0	11.0	0.5	31.8	**25.4**	29.7
[13]	**71.7**	72.8	10.6	53.4	3.1	44.7	64.8	14.6	**2.9**	**52.2**	15.8	37.0
Ours	70.13	**78.14**	**12.18**	**55.26**	9.91	**46.41**	**66.82**	13.53	0.59	46.51	12.17	**37.42**

Table 4. Average Precision (AP) results for HDD Layer 1. While calculating and comparing our **mAP**, AP of **Crossing Vehicle** is not accounted as it was not reported in [10]

	AP results: Layer 1 - Cause Layer						**mAP**
Methods	**Congestion**	**Sign**	**Traffic Light**	**Crossing Vehicle**	**Parked Car**	**Pedestrian**	
[10]	39.72	46.83	45.31	NA	**7.24**	2.15	28.25
Ours	**76.84**	**47.19**	**67.70**	17.42	2.29	**4.54**	**39.71**

the visual and sensor data, at the end of the 5^{th} second). In the experiments $E_{r\times c,16}$, we found that the class-wise accuracy scores for each Scenario plateaus after the 50^{th} epoch, with slight fluctuations later (total number of epochs run - 64). Since the motive of these experiments was to find perceptible differences in the performance of each spatial resolution, we chose not to report the highest class-wise accuracy scores obtained at different epochs; rather we report the results of the 64^{th} epoch for all of the $E_{r\times c,16}$ for a fair comparison. As can be seen in Table 2, scores across different $E_{r\times c,16}$ do not lead to a conclusive evidence about which spatial resolution is superior (we observed similar ambiguity with the highest class-wise accuracy scores too). We attribute this inconclusiveness to the small scale of the dataset; each Scenario is just 1 hour to 3 hours in total. However, it is observable that our 3 FPS visual only 3D-CNNs based model outperforms the 30 FPS multi-modal pre-trained CNN-LSTM based model for few classes, which shows its efficacy.

4.2. **HDD Dataset Results**

Authors in [10, 11, 13] chose to report the Average Precision (AP) scores of each driving scene in a Layer; for comparison, we do the same. We executed the experiments $E_{r\times c,l}$ for both layers, Layer 0: Goal-Oriented Action and Layer 1: Cause, for varying number of epochs (10 to 16). Upon observing the inference mean Average Precision (mAP) scores at the end of each epoch we found that it plateaus (with minimal variations) after 6^{th} epoch in all the experiments. Therefore we present and analyse the AP, mAP, and ASiST@x scores obtained at the end of 7^{th} epoch (this also helps towards fair comparison of different $E_{r\times c,l}$).

4.2.1. **AP and mAP Score Analysis**

Tables 3 and 4 show the AP results of the experiment $E_{108\times 192,16}$ for Layer 0 and Layer 1 respectively. In Table 3, we see that our results vastly outperform the ones [10, 11] obtained by the coupled Convolutional-Recurrent based models. Results of [13] are closer to ours because it corresponds to their *online* C3D framework. It is notable that they [13] obtained their results with input clips of 20 frames and a resolution of 224×224 pixels, whereas, our results with 16 frames clip and $\approx 60\%$ smaller resolution beats theirs in almost half of the driving scenes. In Table 4 we compare our visual only results with [10] obtained on the visual and CAN Bus sensor data (due to the absence of their results on visual data alone). Here also we note that our visual only model beats their multi-modal coupled Convolutional-Recurrent model; thus showing the efficacy of our 3D-CNNs based model to effectively capture the spatiotemporal features.

Algorithm 1: ASiST@x metric calculation

Input : K, Array $[l_i^t]$, and $[l_i^p]$
Output: $lmatched\_at\_x$, $fcount\_at\_x$

Initialization:
  $len\_l^t \leftarrow$ Length of $[l_i^t]$
  $x \leftarrow 0$ /* Relative index of next frame since a scene's transition */
  $lmatched\_at\_x \leftarrow [0, \cdots, 0]$ /* Array of zeros of length $K+1$ */
  $fcount\_at\_x \leftarrow [0, \cdots, 0]$ /* Array of zeros of length $K+1$ */
for $i \leftarrow 1$ **to** len_l^t **do**
  if $l_i^t \neq l_{i-1}^t$ **then**
    /* Scene transition detected */
    $x \leftarrow 0$ /* Relative index since scene transition set to 0 */
  end if
  if $x <= K$ **then**
    $fcount\_at\_x[x] \leftarrow fcount\_at\_x[x] + 1$
    if $l_i^p = l_i^t$ **then**
      $lmatched\_at\_x[x] \leftarrow lmatched\_at\_x[x] + 1$
    end if
  end if
  $x \leftarrow x + 1$
end for
return $lmatched\_at\_x, fcount\_at\_x$

4.2.2. **ASiST@x Analysis**

Due to the absence of per-frame accuracy metrics for both Layers of the HDD dataset, we are first to analyse them. To define the **ASiST@x** metric (**A**ccuracy at the $\mathbf{x}^{th}$ frame **Si**nce **S**cene **T**ransition), let us start by denoting the true label and predicted label of each frame f_i as l_i^t and l_i^p respectively. Here, instead of calculating the *conventional* accuracy metric by comparing the aligned true and predicted label of each frame (which apart from being a high level abstract metric, is also not suitable for a heavily imbalanced dataset), we calculate it in the following way. In a sequence of true labels of a driving session, a scene transition occurs at the frame index i (i.e. in frame f_i) when l_i^t is not equal to l_{i-1}^t. Thus, a contiguous sequence of same valued true labels l_m^t where $m \in [i, \cdots, j]$ denotes an ongoing driving scene in the frame sequence $F_{i,j}$ (i.e. the scene starts and ends at the frame index i and j respectively). Let K denote the number of next frames since the index i at which the scene has transitioned. Note that the set of K frames does not include the frame f_i; therefore, after including f_i in the set, the total number of frames in consideration for analysis increases to $K+1$. Also, a scene can be of smaller duration than K (next) frames, i.e. $j < i+K$, therefore, the actual number of scene frames into consideration is $n+1$ where $n = min(j-i, K)$. Next, let us define two zero-valued arrays: $lmatched\_at\_x = [0, \cdots, 0]$ and $fcount\_at\_x = [0, \cdots, 0]$ each of shape $K+1$. At index x, the $lmatched\_at\_x$ array stores the count of occurrences when $l_{i+x}^p = l_{i+x}^t$ and $fcount\_at\_x$ array stores the count of frames (since the scene transition) at index $i+x$ for $x \in [0, \cdots, n]$. Note that if $l_{i+x}^p = l_{i+x}^t$ at $x = 0$, then it implies that the driving scene was correctly recognized right at its apt transition. Also, if $l_{i+x}^p = l_{i+x}^t$ at $x > 0$, it is possible that the scene was recognized first at an earlier index in range $[i, \cdots, i+x-1]$. For a session, we define $\mathbf{ASiST@x} = \frac{lmatched\_at\_x}{fcount\_at\_x}$ (element wise division) after computing the $lmatched\_at\_x$ and $fcount\_at\_x$ for its scenes. Thus, the metric ASiST@x not only measures the efficacy

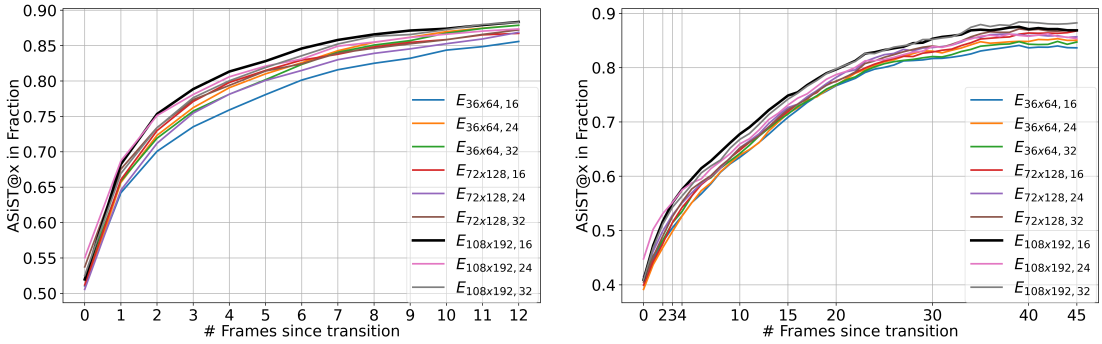
Table 5. Influence of the governing elements on ASiST@x curve form. Note: Above analysis is subject to variability due to the stochasticity of scenes duration and the value of k.

Elements governing the ASiST@x curve form			**Effect on ASiST@x curve form** $x \in [0, \cdots, K]$
Scene duration compared to K + 1 frames	**x^{th} frame at which the scene is recognized first**	**Recognition continuity: Continuous or Irregular?**	
All $< K+1$	All scenes recognized right at $x = 0$	Continuous	ASiST@x = 100% $\forall$ x up to a certain value, then the curve drops to 0
—"—	—"—	Irregular after $x = k$	ASiST@x = 100% $\forall$ $x \in [0, \cdots, k]$, then curve wiggles and drops to 0
—"—	Scenes recognized at $x \geq 0$, some not recognized at all	Continuous	ASiST@x $< 100\%$ at $x = 0$, then curve rises (may wiggle and reach 100% if individual duration vary), then drops to 0
—"—	—"—	Irregular after $x = k$	ASiST@x $< 100\%$ at $x = 0$ then curve rises (may wiggle and reach 100% if individual duration vary) then wiggles, and drops to 0
Some (or None) $<$ $K+1$, rest $\geq K+1$	All scenes recognized right at x=0	Continuous	ASiST@x = 100% $\forall$ x
—"—	—"—	Irregular after $x = k$	ASiST@x = 100% $\forall$ $x \in [0, \cdots, k]$, then curve wiggles
—"—	Scenes recognized at $x \geq 0$, some not recognized at all	Continuous	ASiST@x $< 100\%$ at $x = 0$, then curve rises (may wiggle and reach 100% if few scenes' duration $< K+1$), and peaks and may plateau/wiggle/fall slightly
—"—	—"—	Irregular after $x = k$	ASiST@x $< 100\%$ at $x = 0$, then curve rises up to $x = k$, but does not reach 100% if $k <$ all scenes' duration (otherwise may reach 100%), and then wiggles

of a model to recognize a scene at the x^{th} frame after its true transition, but also implicitly measures the continuity of recognizing an occurring scene. In other words, ASiST@x scores tell us the percentage of scenes (that are at least $x+1$ frames long) that have been recognized by the model by the arrival of the x^{th} frame since its transition.

In **Algorithm 1** we present an efficient implementation for calculating the ASiST@x metric. After obtaining the $lmatched\_at\_x$ and $fcount\_at\_x$ arrays for each of the test sessions in the HDD dataset, we calculate the overall ASiST@x $= \frac{\sum_{\text{session}} lmatched\_at\_x}{\sum_{\text{session}} fcount\_at\_x}$ (element wise summation, element wise division). Note that during ASiST@x scores calculation, the scene transition from an ongoing event (e.g. left turn) to the background class (i.e. no ongoing event) is also considered. It can be inferred from our definition of the ASiST@x metric that it is governed by three elements, namely: (1) Duration of the scenes, (2) x^{th} frame at which the scene transition is recognized, and (3) Recognition continuity of an occurring scene. The Table 5 details down the effect of these three governing elements on the curve form of the ASiST@x metric. It is apparent from the analysis in Table 5 that in general setting, the curves should ideally plateau at a high value as early as possible (with respect to x). Figure 2a and Figure 2b correspond to the ASiST@x scores for Layers 0 and 1 respectively. We set $K = 12$ for Layer 0 and $K = 45$ for Layer 1 since our frame sampling rate is set to 3 FPS and we found the mean duration of the scenes in the respective layers to be 3.82s (std: 2.77s) and 14.57s (std: 18.05s).

In Fig. 2a we see that at $x = 0$, $E_{108\times192,24}$ outperforms others by recognizing 55.02% of the Layer 0 driving scenes right at their true transition. At $x = 1$, this combination of spatial and temporal resolution again outperforms others by recognizing a scene transition 1 frame later (than its true transition) or 1 frame into its predicted transition (i.e. at the



(a) HDD Layer 0: Goal-Oriented Action *(b)* HDD Layer 1: Cause

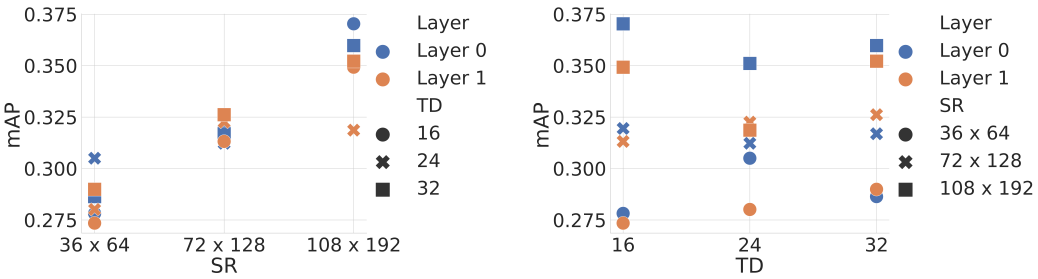
Figure 2. **ASiST@x** plots for HDD dataset layers. Line corresponding to the best performing combination is in black. Best viewed in color.

next frame if the scene transition was already recognized at $x = 0$) in 68.71% of the scenes ($E_{108\times192,16}$ performs nearly same - 68.24%). However, for $x >= 2$, $E_{108\times192,16}$ emerges as the clear winner. Considering its ASiST@3 score, our model takes only 1 second of time (since the true start of scenes) to correctly recognize a scene transition or an occurring scene (if the scene transition was recognized earlier) in 78.86% of all Layer 0 scenes. With respect to its ASiST@4 score, this combination of spatial and temporal resolution again enables our model to detect a scene transition at or earlier than 1.33 seconds (since true start) in 81.34% of all Layer 0 scenes which are at least 5 frames long (@ 3 FPS). We see that ASiST@x curves for all $E_{r\times c,l}$ keep rising, do not wiggle perceptibly, and do not fall within the range of K. This relates strongly to the analysis present in second last row of Table 5, thus hinting towards our model's ability ($\forall\ E_{r\times c,l}$) to continually recognize an occurring scene (after recognizing its transition earlier) apart from recognizing few extra scene transitions in the later frames. With respect to the scenes duration, we found that 35.41% of all scenes (in Layer 0 test data) had window size $< K + 1$ and 25.78% had window size < 10 frames.

In Fig. 2b, we see similar ASiST@x curve form for Layer 1 driving scenes, except that the curves are not as smooth as those in Fig. 2a. Considering the ASiST@0 score, we again see that $E_{108\times192,24}$ outperforms all others by achieving 44.78% score, but soon $E_{108\times192,16}$ takes over (at $x = 3$) and outperforms rest at higher values of x. From ASiST@3 score of $E_{108\times192,16}$ we see that it is able to recognize a scene transition at or earlier than 1s in 55.12% of all Layer 1 scenes which are at least 4 frames long (@ 3 FPS). We also see that $E_{108\times192,32}$ strongly contends with $E_{108\times192,16}$ at values of $x > 15$. This hints that increased temporal context might be necessary for recognizing longer duration scenes (recollect that mean duration of Layer 1 scenes is $14.57s$), but since the Layer 1 scenes' duration are highly variable (std: $18.05s$), it cannot be conclusively established. The ASiST@x curves for all $E_{r\times c,l}$ keep rising and seemingly plateau at values of x closer to K and perceptibly wiggle too. This wiggliness can be attributed to one or more of the following reasons: (1) it can be an image artefact due to the packing of comparatively (with respect to Figure 2a) large number of frames-since-transition, (2) it can be due to the highly variable duration of Layer 1 scenes - implying many scenes end earlier, (3) it can be due to the possible discontinuity in recognizing ongoing scenes. With respect to scenes duration, we found 44.95% and 33.72% of scenes (in Layer 1 test data) which were smaller than $K + 1$ and 30 frames respectively.

4.2.3. **Effect Analysis of Spatial Resolution and Temporal Depth**

Here we study the individual effects of spatial resolution and temporal depth variation on mAP while keeping the other constant. Figure 3a shows an increasing trend in the mAP scores for both Layers and different temporal depths as the spatial resolution increases. This suggests using higher spatial resolution inputs but it comes with an increased cost of



(a) Varying Spatial Resolution (SR) *(b)* Varying Temporal Depth (TD)

Figure 3. Individual effect of varying the Spatial Resolution and Temporal Depth. Layer 0 and Layer 1 are color coded. Best viewed in color.

computational requirements. Also, we found in Section 4.2.1 that a resolution of 108×192 pixels suffices the performance of 224×224 pixels resolution. Upon observing the class-wise AP scores, we found that increasing spatial resolution helps in better recognition of Left Lane Change, Right Lane Change, Left Lane Branch, Sign, Traffic Light, and Crossing Vehicle. In Figure 3b we do not see a definitive trend in the mAP scores as the temporal depth increases. Upon observing the class-wise AP scores for certain fixed spatial resolutions, we found that increased temporal depths resulted slightly better performance for recognizing U-turn, Right Lane Change, Left Lane Change, and Crossing Vehicle. This subtly suggests that the determination of an ideal temporal depth is driving scene dependent, however this hypothesis requires further investigation. Since the mAP (in Figure 3b) does not strongly depend on the temporal depth, and the mAP scores with 16 frames temporal depth is higher or comparable to others, we favour the 16 frames temporal depth due to lesser computations. Conclusively, it can be said that a combination of 108×192 pixels spatial resolution and 16 frames temporal depth performs best for recognizing real time driving scenes in general setting.

5. Conclusion

From our extensively conducted experiments, we showed the success of our C3D inspired 3D-CNNs based model for Driving Scene Understanding. Our visuals only model was found to be comparable and outperformed a variety of visual and multi-modal Driving Scene Understanding approaches as seen in Tables 2, 3, and 4. In accordance with our intention to determining the scene transitions right at their onset, we introduced the ASiST@x metric to evaluate the efficacy of our approach. This metric can be extended to other datasets and different types of tasks as well which deal with the problem of continuous activity/scene recognition. We found that our model achieves ASiST@0 scores of 55.02% and 44.78% for Layers 0 and 1 of the HDD dataset respectively with an input resolution of 108×192 pixels and a temporal depth of 24 frames. In addition, our model takes just 1s of time (at 3 FPS) since the true start of scenes to correctly recognize scene transitions in 78.86% and 55.12% of Layer 0 and Layer 1 driving scenes respectively. We experimentally found the combination of 108×192 pixels resolution and 16 frames temporal depth to be the best among other combinations for recognizing real time driving scenes in the largest real world public dataset. Owing to the demonstrated success of 3D-CNNs, we surmise that creation of multi-modal frameworks (to incorporate e.g. CAN Bus sensor data) with 3D-CNNs would push the results further. One can also explore increasing the Convolutional kernel size and developing shallower architectures to keep the number of trainable parameters in check, thereby leveraging higher spatial resolutions and examining the effect of shorter temporal depths. In addition, one may also segregate the HDD dataset scenes into groups with sufficiently varying means and low standard deviation (of scenes duration) to study the effect of temporal depths in detail, apart from developing more explicit metrics for detecting

continuity in driving scene recognition. Finally, we hope that our insights in the necessary spatial resolution and temporal depth serve as the initial considerations when researchers toil over choosing these hyper-parameters.

Acknowledgements

We would like to thank Yi-Ting Chen [10] for promptly helping us navigate through the HDD dataset. This work is supported by the NSERC Grant Award AWD-013766.

References

[1] H. Berndt, J. Emmert, and K. Dietmayer. "Continuous driver intention recognition with hidden markov models". In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE. 2008, pp. 1189–1194.

[2] H. Hou, L. Jin, Q. Niu, Y. Sun, and M. Lu. "Driver intention recognition method using continuous hidden Markov model". In: *International Journal of Computational Intelligence Systems* 4.3 (2011), pp. 386–393.

[3] J. Ding, R. Dang, J. Wang, and K. Li. "Driver intention recognition method based on comprehensive lane-change environment assessment". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 214–220.

[4] S. Casas, W. Luo, and R. Urtasun. "Intentnet: Learning to predict intention from raw sensor data". In: *Conference on Robot Learning*. 2018, pp. 947–956.

[5] D. Frossard, E. Kee, and R. Urtasun. "DeepSignals: Predicting Intent of Drivers Through Visual Signals". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9697–9703.

[6] A. T. Schulz and R. Stiefelhagen. "Pedestrian intention recognition using latent-dynamic conditional random fields". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2015, pp. 622–627.

[7] H. Bi, R. Zhang, T. Mao, Z. Deng, and Z. Wang. "How Can I See My Future? FvTraj: Using First-Person View for Pedestrian Trajectory Prediction". In: *European Conference on Computer Vision*. Springer. 2020, pp. 576–593.

[8] M. Torstensson, B. Duran, and C. Englund. "Using recurrent neural networks for action and intention recognition of car drivers". In: *8th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2019, Prague, Czech Republic; 19-21 February, 2019*. SciTePress. 2019, pp. 232–242.

[9] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. "VIENA^2: A Driving Anticipation Dataset". In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 449–466.

[10] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko. "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7699–7707.

[11] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall. "Temporal recurrent networks for online action detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 5532–5541.

[12] C. Li, S. H. Chan, and Y.-T. Chen. "Who Make Drivers Stop? Towards Driver-centric Risk Assessment: Risk Object Identification via Causal Inference". In: *arXiv preprint arXiv:2003.02425* (2020).

[13] C. Li, Y. Meng, S. H. Chan, and Y.-T. Chen. "Learning 3d-aware egocentric spatial-temporal interaction via graph convolutional networks". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8418–8424.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[15] A. Diba, A. M. Pazandeh, and L. Van Gool. "Efficient two-stream motion and appearance 3d cnns for video classification". In: *arXiv preprint arXiv:1608.08851* (2016).

[16] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool. "Temporal 3d convnets: New architecture and transfer learning for video classification". In: *arXiv preprint arXiv:1711.08200* (2017).

[17] S. Ji, W. Xu, M. Yang, and K. Yu. “3D convolutional neural networks for human action recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), pp. 221–231.

[18] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.